

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Webcam

##### 2.1.1 Pengertian Webcam

*Webcam* (singkatan dari kamera web) adalah sebutan bagi kamera waktu-nyata yang gambarnya bisa dilihat melalui *www (World Wide Web)*, program pengolah pesan cepat, atau aplikasi pemanggilan video. Istilah webcam merujuk pada teknologi secara umumnya, sehingga kata *webcam* kadang-kadang diganti dengan kata lain yang memberikan pemandangan yang ditampilkan di kamera. Kamera web dapat diartikan juga sebagai sebuah kamera video digital kecil yang dihubungkan ke komputer melalui *port USB*, *port COM* atau dengan jaringan *Ethernet* atau *Wi-Fi*. Contoh *webcam* dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Webcam

##### 2.1.2 Fungsi Webcam

Fungsi dari *webcam* telah kita ketahui yaitu untuk memudahkan kita dalam mengolah pesan cepat seperti *chat* melalui video atau bertatap muka melalui video secara langsung. *Webcam* juga berfungsi sebagai alat untuk *men-transfer* sebuah media secara langsung, namun perlu disadari kebanyakan pengguna menggunakan piranti ini hanya untuk *chat video*.



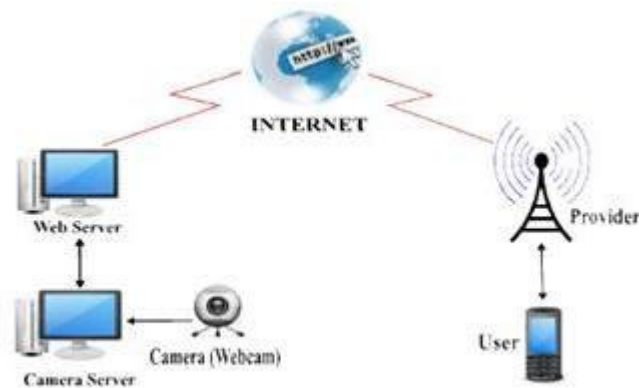
### 2.1.3 Cara Kerja Webcam

Sebuah *webcam* yang sederhana terdiri dari: sebuah lensa standar, dipasang di sebuah papan sirkuit untuk menangkap sinyal gambar, *casing (cover)*, termasuk *casing* depan dan *casing* samping untuk menutupi lensa standar dan memiliki sebuah lubang lensa di *casing* depan yang berguna untuk memasukkan gambar, kabel *support* yang dibuat dari bahan yang *fleksibel*, salah satu ujungnya dihubungkan dengan papan sirkuit dan ujung satu lagi memiliki *connector*, kabel ini dikontrol untuk menyesuaikan ketinggian, arah dan sudut pandang kamera web. Sebuah *webcam* biasanya dilengkapi dengan *software*, *software* ini mengambil gambar-gambar dari kamera digital secara terus menerus ataupun dalam *interval* waktu tertentu dan menyiarkannya melalui koneksi internet. Ada beberapa metode penyiaran, metode yang paling umum adalah hardware mengubah gambar ke dalam bentuk *file JPG* dan menguploadnya ke *web server* menggunakan *File Transfer Protocol (FTP)*.

*Frame rate* mengindikasikan jumlah gambar sebuah *software* dapat ambil dan *transfer* dalam satu detik. Untuk *streaming* video, dibutuhkan minimal 15 *frame per second (fps)* atau idealnya 30 fps. Untuk mendapatkan *frame rate* yang tinggi, dibutuhkan koneksi internet yang tinggi kecepatannya. Sebuah kamera web tidak harus selalu terhubung dengan komputer, ada kamera web yang memiliki *software webcam* dan *web server built-in*, sehingga yang diperlukan hanyalah koneksi internet. Kamera web seperti ini dinamakan *network camera*. Kita juga bisa menghindari penggunaan kabel dengan menggunakan hubungan radio, koneksi *Ethernet* ataupun *WiFi*.

Pada perancangan aplikasi *webcam* untuk digunakan pada mikroskop digital, *webcam* akan dihubungkan ke PC untuk menampilkan hasil gambar yang terlihat. Untuk mengatur pencahayaan dan fokus, *webcam* tersebut dibantu dengan *arduino uno*. *Webcam* berfungsi sebagai lensa yang digunakan di mikroskop digital. Cara kerja webcam dapat dilihat seperti gambar dibawah ini:

---



**Gambar 2.2** Cara kerja webcam secara umum

(Sumber : [www.wikipedia.com/Web\\_Camera.html](http://www.wikipedia.com/Web_Camera.html))

## 2.2 Laptop

Laptop adalah komputer bergerak (bisa dipindahkan dengan mudah) yang berukuran relatif kecil dan ringan, beratnya berkisar dari 1-6 kg, tergantung ukuran, bahan, dari spesifikasinya, laptop dapat digunakan dalam lingkungan yang berbeda dari komputer. Laptop termasuk layar, *keyboard*, dan *trackpad* atau *trackball*, yang berfungsi sebagai *mouse*. Karena dimaksudkan untuk digunakan di mana saja, laptop memiliki baterai yang memungkinkan untuk beroperasi tanpa terhubung ke sumber listrik. Laptop juga termasuk adaptor daya yang memungkinkan untuk menggunakan daya dari stop-kontak dan mengisi kembali baterai.

Laptop secara signifikan lebih lambat dari pada *personal computer*. tetapi kemajuan teknologi manufaktur telah memungkinkan laptop melakukan hampir sama dengan PC (*personal computer*). Bahkan laptop *high-end* sering melakukan lebih baik dari pada *personal computer* yang mempunyai spesifikasi rendah. Kebanyakan laptop juga mencakup beberapa *I/O port*, seperti *USB port*, yang memungkinkan *keyboard standar* dan *mouse* untuk digunakan dengan laptop. Laptop *modern* sering termasuk adaptor jaringan *nirkabel/wireless*, yang



memungkinkan pengguna untuk mengakses *internet* tanpa memerlukan kabel. Gambar 2.3 merupakan contoh dari laptop.



**Gambar 2.3** NoteBook

(Sumber : [www.rozetka.com/asus\\_s200.html](http://www.rozetka.com/asus_s200.html))

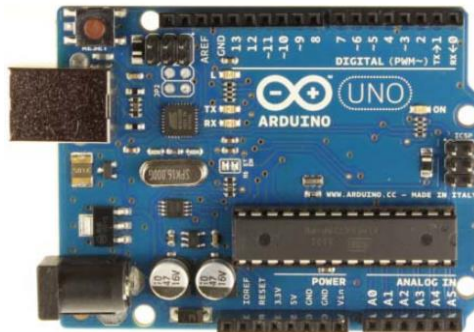
### **2.3 Arduino Uno**

Arduino Uno adalah arduino board yang menggunakan mikrokontroler ATmega328. Arduino Uno memiliki 14 pin digital (6 pin dapat digunakan sebagai output PWM), 6 input analog, sebuah 16 MHz osilator kristal, sebuah koneksi USB, sebuah konektor sumber tegangan, sebuah *header* ICSP, dan sebuah tombol reset. Arduino Uno memuat segala hal yang dibutuhkan untuk mendukung sebuah mikrokontroler. Hanya dengan menghubungkannya ke sebuah komputer melalui *USB* atau memberikan tegangan DC dari baterai atau *adaptor* AC ke DC sudah dapat membuanya bekerja. Arduino Uno menggunakan ATmega16U2 yang diprogram sebagai *USB-to-serial converter* untuk komunikasi serial ke komputer melalui port USB.

"Uno" berarti satu di Italia dan diberi nama untuk menandai peluncuran Arduino 1.0. Versi 1.0 menjadi versi referensi Arduino ke depannya. Arduino Uno R3 adalah revisi terbaru dari serangkaian board Arduino, dan model referensi



untuk platform Arduino. Tampak atas dari arduino uno dapat dilihat pada Gambar 2.4.



**Gambar 2.4** Arduino Uno

(Sumber: repository.usu.ac.id/bitstream/123456789/37482/4/Chapter%20II.pdf)

Adapun data teknis board Arduino UNO R3 adalah sebagai berikut:

- Mikrokontroler : ATmega328
- Tegangan Operasi : 5V
- Tegangan *Input (recommended)* : 7 - 12 V
- Tegangan *Input (limit)* : 6-20 V
- Pin digital *I/O* : 14 (6 diantaranya pin *PWM*)
- Pin Analog *input* : 6
- Arus DC per pin *I/O* : 40 mA
- Arus DC untuk pin 3.3 V : 150 mA
- *Flash Memory* : 32 KB dengan 0.5 KB digunakan untuk *bootloader*
- *EEPROM* : 1 KB
- Kecepatan Pewaktuan : 16 Mhz

### 2.3.1 Pin Masukan dan Keluaran Arduino Uno

Masing-masing dari 14 pin digital arduino uno dapat digunakan sebagai masukan atau keluaran menggunakan fungsi *pinMode()*, *digitalWrite()* dan *digitalRead()*. Setiap pin beroperasi pada tegangan 5 volt. Setiap pin mampu menerima atau menghasilkan arus maksimum sebesar 40 mA dan memiliki 10



resistor *pull-up internal* (diputus secara *default*) sebesar 20-30 KOhm. Sebagai tambahan, beberapa pin masukan digital memiliki kegunaan khusus yaitu:

- Komunikasi serial: pin 0 (RX) dan pin 1 (TX), digunakan untuk menerima (RX) dan mengirim (TX) data secara serial.
- *External Interrupt*: pin 2 dan pin 3, pin ini dapat dikonfigurasi untuk memicu sebuah interrupt pada nilai rendah, sisi naik atau turun, atau pada saat terjadi perubahan nilai.
- *Pulse-width modulation* (PWM): pin 3, 5, 6, 9, 10 dan 11, menyediakan keluaran PWM 8-bit dengan menggunakan fungsi `analogWrite()`.
- *Serial Peripheral Interface* (SPI): pin 10 (SS), 11 (MOSI), 12 (MISO) dan 13 (SCK), pin ini mendukung komunikasi SPI dengan menggunakan *SPI library*.
- *LED*: pin 13, terdapat built-in *LED* yang terhubung ke pin digital 13. Ketika pin bernilai *High* maka *LED* menyala, sebaliknya ketika pin bernilai *Low* maka *LED* akan padam.

Arduino Uno memiliki 6 masukan analog yang diberi label A0 sampai A5, setiap pin menyediakan resolusi sebanyak 10 bit (1024 nilai yang berbeda). Secara *default* pin mengukur nilai tegangan dari *ground* (0V) hingga 5V, walaupun begitu dimungkinkan untuk mengganti nilai batas atas dengan menggunakan pin *AREF* dan fungsi `analogReference()`. Sebagai tambahan beberapa pin masukan analog memiliki fungsi khusus yaitu pin A4 (SDA) dan pin A5 (SCL) yang digunakan untuk komunikasi *Two Wire Interface* (TWI) atau *Inter Integrated Circuit* (I2C) dengan menggunakan *Wire library*.

- *TWI*: A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI.
  - *Aref*. Referensi tegangan untuk input analog. Digunakan dengan `analogReference()`.
  - *Reset*.
-



### 2.3.2 Catu Daya

Arduino uno dapat diberi daya melalui koneksi *USB (Universal Serial Bus)* atau melalui *power supply* eksternal. Jika arduino uno dihubungkan ke kedua sumber daya tersebut secara bersamaan maka arduino uno akan memilih salah satu sumber daya secara otomatis untuk digunakan. *Power supply* eksternal (yang bukan melalui *USB*) dapat berasal dari *adaptor* AC ke DC atau baterai. *Adaptor* dapat dihubungkan ke soket power pada arduino uno. Jika menggunakan baterai, ujung kabel yang dibubungkan ke baterai dimasukkan kedalam pin *GND* dan *Vin* yang berada pada konektor power.

Arduino uno dapat beroperasi pada tegangan 6 sampai 20 volt. Jika arduino uno diberi tegangan di bawah 7 volt, maka pin 5V pada board arduino akan menyediakan tegangan di bawah 5 volt dan mengakibatkan arduino uno mungkin bekerja tidak stabil. Jika diberikan tegangan melebihi 12 volt, penstabil tegangan kemungkinan akan menjadi terlalu panas dan merusak arduino uno. Tegangan rekomendasi yang diberikan ke arduino uno berkisar antara 7-12 volt.

Pin-pin catu daya adalah sebagai berikut:

- Vin adalah pin untuk mengalirkan sumber tegangan ke arduino uno ketika menggunakan sumber daya eksternal (selain dari koneksi *USB* atau sumber daya yang teregulasi lainnya). Sumber tegangan juga dapat disediakan melalui pin ini jika sumber daya yang digunakan untuk arduino uno dialirkan melalui soket power.
  - 5V adalah pin yang menyediakan tegangan teregulasi sebesar 5 volt berasal dari regulator tegangan pada arduino uno.
  - 3V3 adalah pin yang menyediakan tegangan teregulasi sebesar 3,3 volt berasal dari regulator tegangan pada arduino uno.
  - GND adalah pin ground.
-



### 2.3.3 Memori

Arduino Uno adalah arduino board yang menggunakan mikrokontroler ATmega328. Maka peta memori arduino uno sama dengan peta memori pada mikrokontroler ATmega328.

ATmega328 ini memiliki 32 KB dengan 0,5 KB digunakan untuk loading file. Ia juga memiliki 2 KB dari SRAM dan 1 KB dari EEPROM.

#### 2.3.3.1 Memori Data

Memori data ATmega328 terbagi menjadi 4 bagian, yaitu 32 lokasi untuk register umum, 64 lokasi untuk register I/O, 160 lokasi untuk register I/O tambahan dan sisanya 2048 lokasi untuk data SRAM internal. Register umum menempati alamat data terbawah, yaitu 0x0000 sampai 0x001F. Register I/O menempati 64 alamat berikutnya mulai dari 0x0020 hingga 0x005F. Register I/O tambahan menempati 160 alamat berikutnya mulai dari 0x0060 hingga 0x00FF. Sisa alamat berikutnya mulai dari 0x0100 hingga 0x08FF digunakan untuk SRAM internal. Spesifikasi Arduino Uno dapat dilihat Tabel 2.1 berikut :

#### 2.3.3.2 Memori Data EEPROM

Arduino uno terdiri dari 1 KByte memori data EEPROM. Pada memori EEPROM, data dapat ditulis/dibaca kembali dan ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat nonvolatile. Alamat EEPROM dimulai dari 0x000 hingga 0x3FF.

**Tabel 2.1** Deskripsi Arduino Uno

<i>Mikrokontroller</i>	Atmega328
<i>Operasi Voltage</i>	5V
<i>Input Voltage</i>	7-12 V (Rekomendasi)
<i>Input Voltage</i>	6-20 V (limits)
<i>I/O</i>	14 pin (6 pin untuk PWM)





Arus	50 mA
Flash Memory	32KB
Bootloader	SRAM 2 KB
EEPROM	1 KB
Kecepatan	16 Mhz

(Sumber : repository.usu.ac.id/bitstream/123456789/37482/4/Chapter%20II.pdf)

#### 2.3.4 Komunikasi

Arduino uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Firmware Arduino menggunakan *USB driver* standar *COM*, dan tidak ada driver eksternal yang dibutuhkan. Namun pada sistem operasi Windows, format file Inf diperlukan. USB driver Arduino dapat dilihat Gambar 2.4 .

Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke board Arduino. *RX* dan *TX LED* di board akan berkedip ketika data sedang dikirim melalui *chip USB-to-serial* dan koneksi *USB* ke komputer. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi *inteface* pada sistem.



**Gambar 2.5** Kabel USB Arduino Uno

(Sumber: repository.usu.ac.id/bitstream/123456789/37482/4/Chapter%20II.pdf)



### 2.3.5 Programming

Arduino Uno dapat diprogram dengan perangkat lunak Arduino IDE. Pilih Arduino Uno dari *Tool* lalu sesuaikan dengan mikrokontroler yang digunakan. Pada ATmega328 pada Uno Arduino memiliki *bootloader* yang memungkinkan Anda untuk meng-*upload* program baru untuk itu tanpa menggunakan programmer *hardware* eksternal. Arduino uno menggunakan protokol dari Bahasa pemrograman C. Sistem dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau *programmer* DFU (Mac OS dan Linux) untuk memuat *firmware* baru, atau anda dapat menggunakan *header ISP* dengan *programmer* eksternal.

## 2.4 Motor Servo

Motor servo adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor servo merupakan salah satu jenis motor DC.

Seperti namanya, Motor Servo adalah sebuah servo. Lebih khusus lagi adalah servo loop tertutup yang menggunakan umpan balik posisi untuk mengontrol gerakan dan posisi akhir. Masukan kontrolnya adalah beberapa sinyal, baik analog atau digital, yang mewakili posisi yang diperintahkan untuk poros output. Bentuk dari motor servo dapat dilihat pada Gambar 2.6.



**Gambar 2.6** Motor Servo

(Sumber: [elisa.ugm.ac.id/serv\\_.jpg](http://elisa.ugm.ac.id/serv_.jpg))



Servo disusun dari sebuah motor DC, *gearbox*, *variabel resistor* (VR) atau *potensiometer* dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang pada pin kontrol motor servo. Dapat kita lihat pada Gambar 2.7 dibawah ini merupakan konstruksi dari motor servo.



**Gambar 2.7** Konstruksi Motor Servo

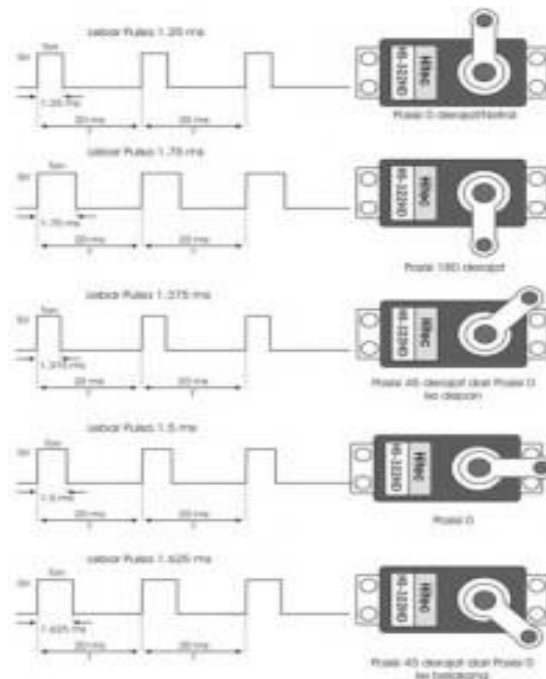
(Sumber : [www.elektronikadasar.co.id/servo\\_motor.html](http://www.elektronikadasar.co.id/servo_motor.html))

Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan dengan memberikan variasi lebar pulsa (*duty cycle*) sinyal PWM pada bagian pin kontrolnya. Adapun jenis-jenis motor servo ada 2, yaitu:

- Motor Servo Standar 180° Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°.
- Motor Servo *Continuous* Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).



Pulsa kontrol Motor Servo Operasional dikendalikan oleh sebuah pulsa selebar  $\pm 20$  ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Apabila motor servo diberikan pulsa dengan besar 1.5 ms mencapai gerakan  $90^\circ$ , maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati  $0^\circ$  dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati  $180^\circ$ , seperti yang terlihat pada Gambar 2.8 dibawah ini



**Gambar 2.8** Pulsa Kendali Motor Servo

(Sumber : [www.elektronikadasar.co.id/pengontrolan\\_servo.html](http://www.elektronikadasar.co.id/pengontrolan_servo.html))

Pulsa Kendali Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi *ton duty cycle* 1.5 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut  $0^\circ$ / netral). Pada saat *ton duty cycle* dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar ke arah yang berlawanan jarum jam (*Counter Clock wise*, CCW) dengan membentuk sudut yang besarnya linier terhadap besarnya *ton duty cycle*,

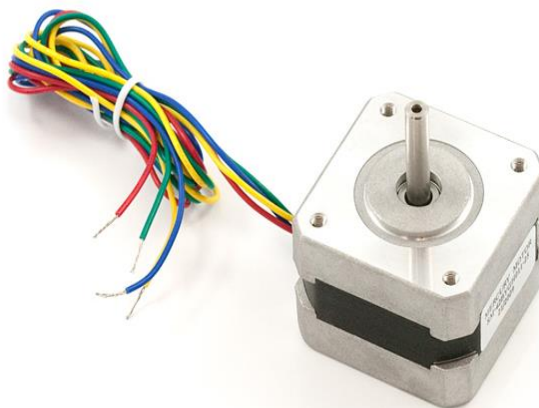


dan akan bertahan diposisi tersebut. Dan sebaliknya, jika *ton duty cycle* dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (Clock Wise, CW) dengan membentuk sudut yang linier pula terhadap besarnya *Ton duty cycle*, dan bertahan diposisi tersebut. (Sumber: [www.elektronikadasar.co.id](http://www.elektronikadasar.co.id))

## 2.5 Motor Stepper

Motor *Stepper* adalah suatu motor listrik yang dapat mengubah pulsa listrik yang diberikan menjadi gerakan motor *discrete* (terputus) yang disebut *step* (langkah). Satu putaran motor memerlukan  $360^\circ$  dengan jumlah langkah yang tertentu per-derajatnya. Ukuran kerja dari motor stepper biasanya diberikan dalam jumlah langkah per-putaran per-detik.

Motor stepper bekerja berdasarkan pulsa-pulsa yang diberikan pada lilitan fasenya dalam urutan yang tepat. Selain itu, pulsa-pulsa itu harus juga menyediakan arus yang cukup besar pada lilitan fase tersebut. Karena itu untuk pengoperasian motor stepper pertama-tama harus mendesain suatu *sequencer logic* untuk menentukan urutan pencatuan lilitan fase motor dan kemudian menggunakan suatu penggerak (*driver*) untuk menyediakan arus yang dibutuhkan oleh lilitan fase. Berikut merupakan contoh motor servo pada Gambar 2.9.



**Gambar 2.9** Motor Stepper

(<https://www.coolcomponents.co.uk/media/catalog/product/0/9/09238-1.jpg>)



Motor *stepper* memiliki pin-pin input yang menjadi kutub-kutub magnet di dalam motor. Bila salah satu pin diberi sumber tegangan, pin tersebut akan mengaktifkan kutub di dalam magnet sebagai kutub utara dan kutub yang tidak diberi tegangan sebagai kutub selatan. Dengan terdapatnya 2 kutub di dalam motor ini, rotor di dalam motor yang memiliki kutub permanen akan mengarah sesuai dengan kutub-kutub input. Kutub utara rotor akan mengarah ke kutub selatan stator sedangkan kutub selatan rotor akan mengarah ke kutub utara stator.

Pada motor *stepper* terdapat berbagai macam tipe, antara lain:

1. *Unipolar*
2. *Bipolar*
3. *Single-phase*
4. *Multi-phase*

Skema umum motor *stepper* dapat dilihat seperti Gambar 2.10.



**Gambar 2.10** Skema Motor *Stepper*

(<http://elektronika-dasar.web.id/teori-elektronika/motor-stepper/>)

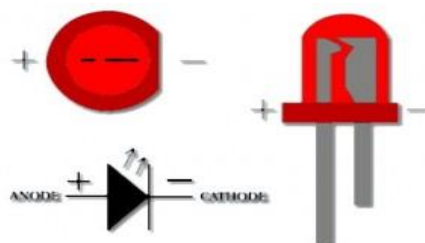
Gambar di atas menunjukkan penampang motor *stepper* dengan empat koil. Setiap koil memiliki empat kondisi kutub. Bila kondisi satu yang aktif, posisi



rotor akan nampak seperti di atas. Bila kondisi bergeser ke dua, rotor akan berputar ke kiri dengan sudut putar sesuai dengan jarak kondisi satu dan dua. Namun bila setelah kondisi satu, kondisi empat yang aktif, rotor akan menuju ke koil dengan pin empat paling dekat dengan pin satu dari kondisi sebelumnya. Hal ini menyebabkan rotor berputar ke kanan dan seterusnya. ketelitian sudut putar pada motor *stepper* sebanding dengan banyaknya kondisi masukkannya. Pada kondisi seperti gambar *stepper* dengan empat koil dan empat kondisi kutub dengan metode *full step* akan mampu menghadap ke 16 sudut berbeda.

## 2.6 LED

LED (*Light Emitting Diode*) adalah dioda yang dapat memancarkan cahaya pada saat mendapat arus bias maju (*forward bias*). LED (*Light Emitting Diode*) dapat memancarkan cahaya karena menggunakan dopping galium, arsenic dan phosphorous. Jenis doping yang berbeda diatas dapat menghasilkan cahaya dengan warna yang berbeda. LED (*Light Emitting Diode*) merupakan salah satu jenis dioda, sehingga hanya akan mengalirkan arus listrik satu arah saja. LED akan memancarkan cahaya apabila diberikan tegangan listrik dengan konfigurasi *forward bias*. Berbeda dengan dioda pada umumnya, kemampuan mengalirkan arus pada LED (*Light Emitting Diode*) cukup rendah yaitu maksimal 20 mA. Apabila LED (*Light Emitting Diode*) dialiri arus lebih besar dari 20 mA maka LED akan rusak, sehingga pada rangkaian LED dipasang sebuah resistor sebagai pembatas arus. Simbol dan bentuk fisik dari LED (*Light Emitting Diode*) dapat dilihat pada Gambar 2.11 berikut.



**Gambar 2.11 LED**

(Sumber : [www.wikipedia.com/LED.html](http://www.wikipedia.com/LED.html))



Dari gambar diatas dapat kita ketahui bahwa LED memiliki 2 buah kaki, yaitu kaki anoda dan kaki katoda. Pada gambar diatas kaki anoda memiliki ciri fisik lebih panjang dari kaki katoda pada saat masih baru, kemudian kaki katoda pada LED (*Light Emitting Diode*) ditandai dengan bagian body LED yang di papas rata. Kaki anoda dan kaki katoda pada LED (*Light Emitting Diode*) disimbolkan seperti pada gambar 2.9 diatas. Pemasangan LED (*Light Emitting Dioda*) agar dapat menyala adalah dengan memberikan tegangan bias maju yaitu dengan memberikan tegangan positif ke kaki anoda dan tegangan negatif ke kaki katoda.

## 2.7 Microsoft Visual C#

C# (dibaca: **C sharp**) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar *ECMA-334 C# Language Specification*, nama C# terdiri atas sebuah huruf Latin **C** (U+0043) yang diikuti oleh tanda pagar yang menandakan angka **#** (U+0023). Tanda pagar **#** yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar **#** (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam *keyboard* standar.

Standar European Computer Manufacturer Association (ECMA) mendaftarkan beberapa tujuan desain dari bahasa pemrograman C#, sebagai berikut:

- a. Bahasa pemrograman C# dibuat sebagai bahasa pemrograman yang bersifat *bahasa pemrograman general-purpose* (untuk tujuan jamak), berorientasi objek, modern, dan sederhana.





- b. Bahasa pemrograman C# ditujukan untuk digunakan dalam mengembangkan komponen perangkat lunak yang mampu mengambil keuntungan dari lingkungan terdistribusi.
- c. Portabilitas *programmer* sangatlah penting, khususnya bagi *programmer* yang telah lama menggunakan bahasa pemrograman C dan C++.
- d. Dukungan untuk internasionalisasi (multi-language) juga sangat penting.
- e. C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem klien-server (*hosted system*) maupun sistem embedded (*embedded system*), mulai dari perangkat lunak yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada perangkat lunak yang sangat kecil yang memiliki fungsi-fungsi terdedikasi.
- f. Meskipun aplikasi C# ditujukan agar bersifat 'ekonomis' dalam hal kebutuhan pemrosesan dan memori komputer, bahasa C# tidak ditujukan untuk bersaing secara langsung dengan kinerja dan ukuran perangkat lunak yang dibuat dengan menggunakan bahasa pemrograman C dan bahasa rakitan.
- g. Bahasa C# harus mencakup pengecekan jenis (*type checking*) yang kuat, pengecekan larik (*array*), pendeteksian terhadap percobaan terhadap penggunaan Variabel-variabel yang belum diinisialisasikan, portabilitas kode sumber, dan pengumpulan sampah (*garbage collection*) secara otomatis.

Contoh program sederhana dalam bahasa Visual C#.Net yang dipakai untuk menghitung jumlah pembayaran dari m\_item buah barang dengan harga m\_price per item ditambah 5% pajak penjualan.

```
double m_item, m_price, tax, total;
m_item = double.Parse(textBox1.Text);
m_price = double.Parse(textBox2.Text);
tax = 0.05;
total = m_item*m_price*( 1 + tax);
```

---



```
label5.Text = total.ToString();  
MessageBox.Show("Well Done.");
```

## 2.8 Bahasa C

Bahasa C adalah bahasa pemrograman yang dapat dikatakan berada di antara bahasa beraras rendah dan beraras tinggi. Bahasa beraras rendah artinya bahasa yang berorientasi pada mesin dan beraras tinggi berorientasi pada manusia. Bahasa beraras rendah, misalnya bahasa assembler, bahasa ini ditulis dengan sandi yang dimengerti oleh mesin saja, oleh karena itu hanya digunakan bagi yang memprogram mikroprosesor. Bahasa beraras rendah merupakan bahasa yang membutuhkan kecermatan yang teliti bagi pemrogram karena perintahnya harus rinci, ditambah lagi masing-masing pabrik mempunyai sandi perintah sendiri. Bahasa tinggi relatif mudah digunakan, karena ditulis dengan bahasa manusia sehingga mudah dimengerti dan tidak tergantung mesinnya. Bahasa beraras tinggi biasanya digunakan pada komputer.

Pencipta bahasa C adalah Brian W. Kernighan dan Denis M. Ritchi, sekitar tahun 1972. Penulisan program dalam bahasa C dilakukan dengan membagi dalam blok-blok, sehingga bahasa C disebut dengan bahasa terstruktur. Bahasa C dapat digunakan di berbagai mesin dengan mudah, mulai dari PC sampai dengan mainframe, dengan berbagai sistem operasi misalnya DOS, UNIX, VMS dan lain-lain.

### 2.8.1 Penulisan Program Bahasa C

Program Bahasa C tidak mengenal aturan penulisan di kolom tertentu, jadi bisa dimulai dari kolom manapun. Namun demikian, untuk mempermudah pembacaan program dan untuk keperluan dokumentasi, sebaiknya penulisan bahasa C diatur sedemikian rupa sehingga mudah dan enak dibaca. Berikut contoh penulisan Program Bahasa C:

---



```
#include <at89c51.h>
main ()
{
.....
.....
}
```

Program dalam bahasa C selalu berbentuk fungsi seperti ditunjukkan dalam main (). Program yang dijalankan berada di dalam tubuh program yang dimulai dengan tanda kurung buka { dan diakhiri dengan tanda kurung tutup }. Semua yang tertulis di dalam tubuh program ini disebut dengan blok.

Tanda () digunakan untuk mengapit argumen suatu fungsi. Argumen adalah suatu nilai yang akan digunakan dalam fungsi tersebut. Dalam fungsi main diatas tidak ada argumen, sehingga tak ada data dalam (). Dalam tubuh fungsi antara tanda { dan tanda } ada sejumlah pernyataan yang merupakan perintah yang harus dikerjakan oleh prosesor. Setiap pernyataan diakhiri dengan tanda titik koma ;

Baris pertama **#include <...>** bukanlah pernyataan, sehingga tak diakhiri dengan tanda titik koma (;). Baris tersebut meminta kompiler untuk menyertakan file yang namanya ada di antara tanda <...> dalam proses kompilasi. File-file ini (ber-ekstensi .h) berisi deklarasi fungsi ataupun variable. File ini disebut **header**. File ini digunakan semacam perpustakaan bagi pernyataan yang ada di tubuh program.

### 2.8.2 Tipe data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh computer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya



bertipe float maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

**Tabel 2.2** Bentuk tipe data

No	Tipe Data	Ukuran	Range (Jangkauan)	Format	Keterangan
1	Char	1 byte	-128 s/d 127	%c	Karakter
2	Int	2 byte	-32768 s/d 32767	%i, %d	Bilangan bulat
3	Float	4 byte	-3.4E-38 s/d 3.4E+38	%f	Bilangan pecahan
4	Double	8 byte	1.7E-308 s/d 1.7E+308	%lf	Pecahan presisi ganda
5	Void	0 byte	-	-	Tidak bertipe

(Sumber : <http://jamilah.staff.gunadarma.ac.id/pengenalan-c-sdcc.pdf>)

### 2.8.3 Konstanta

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta nilainya selalu tetap. Konstanta harus didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai integer, pecahan, karakter dan string.

Contoh konstanta : 50; 13; 3.14; 4.50005; 'A'; 'Bahasa C'.

### 2.8.4 Variabel

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variable bisa diubah-ubah sesuai kebutuhan. Nama dari suatu variable dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut :

- Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C bersifat case-sensitive artinya huruf besar dan kecil dianggap berbeda.



- Tidak boleh mengandung spasi
- Tidak boleh mengandung symbol-simbol khusus, kecuali garis bawah (underscore). Yang termasuk symbol khusus yang tidak diperbolehkan antara lain : \$, ?, %, #, !, &, \*, (, ), -, +, = dsb
- Panjangnya bebas, tetapi hanya 32 karakter pertama yang terpakai.

### 2.8.5 Deklarasi

Deklarasi diperlukan bila kita akan menggunakan pengenalan (identifikasi) dalam program. Identifikasi dapat berupa variabel, konstanta dan fungsi.

#### 2.8.5.1 Deklarasi Variabel

Bentuk umum pendeklarasian suatu variabel adalah : Nama\_tipe nama\_variabel;

Contoh:

```
int x; // Deklarasi x bertipe integer
```

```
char y, huruf, nim[10]; // Deklarasi variabel bertipe char
```

```
float nilai; // Deklarasi variabel bertipe float
```

```
double beta; // Deklarasi variabel bertipe double
```

```
int array[5][4]; // Deklarasi array bertipe integer
```

```
char *p; // Deklarasi pointer p bertipe char
```

#### 2.8.5.2 Deklarasi Konstanta

Contohnya :

```
#define PHI 3.14 #define nim "0111500382"
```

```
#define nama "Sri Widhiyanti"
```

#### 2.8.5.3 Deklarasi Fungsi

Fungsi merupakan bagian yang terpisah dari program dan dapat diaktifkan atau dipanggil di manapun di dalam program. Fungsi dalam bahasa

---



C ada yang sudah disediakan sebagai fungsi pustaka seperti `printf()`, `scanf()`, `getch()` dan untuk menggunakannya tidak perlu dideklarasikan.

Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh programmer. Bentuk umum deklarasi sebuah fungsi adalah :

Tipe\_fungsi nama\_fungsi(parameter\_fungsi);

Contohnya:

`float luas_lingkaran(int jari);`

`void tampil();`

`int tambah(int x, int y);`

## 2.8.6 Operator

### 2.8.6.1 Operator Penugasan

Operator Penugasan (*Assignment operator*) dalam bahasa C berupa tanda sama dengan (“=”).

### 2.8.6.2 Operator Aritmatika

Bahasa C menyediakan lima operator aritmatika, yaitu :

- `.*` : untuk perkalian
- `/` : untuk pembagian
- `%` : untuk sisa pembagian (modulus)
- `+` : untuk penjumlahan
- `-` : untuk pengurangan

- Perkalian

Contoh :

`//Program Perkalian`

`#include <at89x51.h>`

`void main()`

`{`

---



```
int bil1,bil2;
bil1=4;
bil2=2;
P1=bil1*bil2; //bil1 dikali bil2 hasilnya
dikeluarkan ke P1
}
```

- Pembagian

Contoh :

```
//Program Pembagian
#include <at89x51.h>
void main()
{
int bil1,bil2;
bil1=10;
bil2=2;
P1=bil1/bil2; //hasil pembagian dikeluarkan ke P1
}
```

- Modulus

Contoh :

```
//Program modulus
#include <at89x51.h>
void main()
{
int bil1,bil2;
bil1=13;
bil2=2;
```



```
P1=bil1%bil2; //hasil modulus dikeluarkan ke P1
}
```

- Penjumlahan

Contoh :

```
//program Penjumlahan
#include <at89x51.h> //deklarasi register at89c51
void main()
{
    char bil1,bil2; //deklarasi variabel bil1 dan bil2
    bil1=0x30; //bil1=0x30 dan bil2=0x20 (bentuk
    heksadesimal)
    bil2=0x20;
    P1=bil1+bil2; //hasil penjumlahan dikeluarkan ke
    Port1
}
```

- Pengurangan

Contoh :

```
//Program Pengurangan
#include <at89x51.h>
void main()
{
    int bil1,bil2;
    bil1=0x30;
    bil2=0x20;
    P1=bil1-bil2; // hasil pengurangan dikeluarkan ke
    P1.
}
```





### 2.8.7 Operator Hubungan

Operator hubungan digunakan untuk membandingkan hubungan antara dua buah operand /sebuah nilai atau variable. Operasi majemuk seperti pada tabel dibawah ini:

**Tabel 2.3** Operator Hubungan

Operator	Arti	Contoh	
<	Kurang dari	$X < Y$	Apakah X kurang dari Y
<=	Kurang dari sama dengan	$X \leq Y$	Apakah X Kurang dari sama dengan Y
>	Lebih dari	$X > Y$	Apakah X Lebih dari Y
>=	Lebih dari sama dengan	$X \geq Y$	Apakah X Lebih dari sama dengan Y
==	Sama dengan	$X == Y$	Apakah X Sama dengan Y
!=	Tidak sama dengan	$X != Y$	Apakah X Tidak sama dengan Y

(Sumber : <http://jamilah.staff.gunadarma.ac.id/pengenalan-c-sdcc.pdf>)

### 2.8.8 Operator Logika

Jika operator hubungan membandingkan hubungan antara dua buah operand, maka operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan.

Operator logika ada tiga macam, yaitu :

- && : Logika AND (DAN)
- || : Logika OR (ATAU)
- ! : Logika NOT (INGKARAN)

Operasi AND akan bernilai benar jika dua ekspresi bernilai benar. Operasi OR akan bernilai benar jika dan hanya jika salah satu ekspresinya bernilai benar.



Sedangkan operasi NOT menghasilkan nilai benar jika ekspresinya bernilai salah, dan akan bernilai salah jika ekspresinya bernilai benar.

Contoh :

```
//Program Operator Logika
#include <at89x51.h>
void main(void)
{
    char in1;
    char in2;
    in1=P2;
    in2=P3;
    if((in1==0xf0) && (in2==0x40))
    {P1 = 0x2A;}
}
```

## 2.9 Mikroskop



**Gambar 2.12** Mikroskop

(<https://id.wikipedia.org/wiki/Mikroskop>)

Mikroskop adalah sebuah alat untuk melihat objek yang terlalu kecil untuk dilihat secara kasat mata. Mikroskop merupakan alat bantu yang dapat ditemukan hampir diseluruh laboratorium untuk dapat mengamati organisme berukuran kecil (mikroskopis).

---



Jenis paling umum dari mikroskop, dan yang pertama diciptakan, adalah mikroskop optis. Mikroskop ini merupakan alat optik yang terdiri dari satu atau lebih lensa yang memproduksi gambar yang diperbesar dari sebuah benda yang diletakkan di bidang fokal dari lensa tersebut.

Ada dua bagian utama yang umumnya menyusun mikroskop, yaitu:

- Bagian optik, yang terdiri dari lensa objektif dan lensa okuler.
- Bagian non-optik, yang terdiri dari kaki dan lengan mikroskop, diafragma, meja objek/meja preparat, pemutar halus dan kasar, penjepit kaca objek (preparat), cermin, kondensor, dan sumber cahaya.